

MÉTODOS NUMÉRICOS

Dr. Ing. Claudio Careglio

Facultad de Ingeniería, Doctorado en Ingeniería

- Ejemplo

- Desarrollar un algoritmo para leer la base de un triángulo, la altura y calcular el área:

```
function area_triang_01
    % Reales (ba; h; A)
    ba = 3;
    h = 6; ;

    A = ba*h/2;
    disp('el Area es:'); disp(A);
end
```

- Otra forma:

```
function area_triang
    % Reales (ba; h; A)
    disp('Ingrese base del triangulo:');
    ba = input('');
    disp('Ingrese altura del triangulo:');
    h = input('');

    A = ba*h/2;
    disp('el Area es:'); disp(A);
end
```

2. Estructura tipo Decisión

- Estructura que permite bifurcar una línea de proceso.
 - Analiza condición lógica:
 - Según sea **verdadero** ó **falso** se realiza uno u otro proceso .

- Pseudocódigo

If (condición) **then**

sentencias 1

Else

sentencias 2

Endif

MATLAB

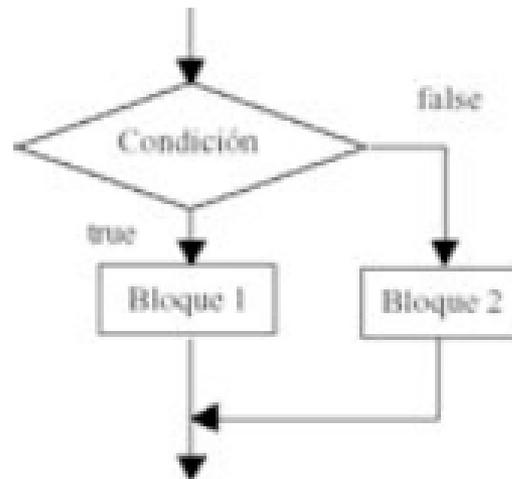
if (condición)

sentencias 1

else

sentencias 2

end



- MATLAB:

- Otras formas según el número de condiciones

- Más simple

if (condición)

sentencias

end

- El visto anteriormente

if (condición)

sentencias 1

else

sentencias 2

end

- Más complejo

if (condición 1)

sentencias 1

elseif (condición 2)

sentencias 2

elseif (condición 3)

sentencias 3

else

sentencias 4

end

2. Estructura tipo Decisión

- Ejemplo
 - Desarrollar un algoritmo para calcular las raíces de la ecuación de segundo grado $ax^2 + bx + c = 0$ y que muestre el resultado:

```
function raices_seg
    % Reales (a; b; c; Discrim; r1;r2)
    disp('Ingrese Coeficiente a:');
    a = input('');
    disp('Ingrese Coeficiente b:');
    b = input('');
    disp('Ingrese Coeficiente c:');
    c = input('');
    Discrim=( b^2-4*a*c)
    if (Discrim > 0)
        r1=(-b+ Discrim^0.5)/(2*a)
        r2=(-b- Discrim^0.5)/(2*a)
        disp( 'Las raíces son:');disp(r1);disp(r2)
    else
        disp( 'Las raíces son complejas conjugadas:')
    end
end
```

3. Estructura tipo Variar

- Estructura que permite repetir una serie de sentencias un **número finito** y **preestablecido** de veces.

- Pseudocódigo

```
Do for variable = comienzo to final step incremento  
    sentencias  
enddo
```

- MATLAB

```
for variable = comienzo:incremento:final  
    sentencias  
end
```

Si "*incremento*" no se especifica, un incremento de 1 es asumido por MATLAB

3.Estructura tipo Variar

- Ejemplo:
 - Desarrollar un código que calcule la suma de dos vectores.

```
clear
```

```
clc
```

```
U=[1; 2; 3]
```

```
V=[10; 10; 10]
```

```
[m,n]=size(U)
```

```
[r,s]=size(V)
```

```
for i=1:m
```

```
    W(i)=U(i)+V(i);
```

```
end
```

```
disp('W='),disp(W)
```

- Ejemplo:
 - Desarrollar un código que calcule la suma de dos matrices.

```
A=[1 2 3;4 5 1]
```

```
B=[1 2 5; 1 3 4]
```

```
[m,n]=size(A);
```

```
[r,s]=size(B);
```

```
for i=1:m
```

```
    for j=1:s
```

```
        C(i,j)=A(i,j)+B(i,j);
```

```
    end
```

```
end
```

```
disp('C='),disp(C)
```

- Ejemplo:

- Desarrollar un código que calcule el producto de dos matrices.
Controlar previamente las dimensiones de las mismas.

```
A=[1 2 3;4 5 1]
```

```
B=[1 2; 1 1; 3 4]
```

```
[m,n]=size(A);
```

```
[r,s]=size(B);
```

```
if (n~=r)
```

```
    error('error en las dimensiones')
```

```
end
```

```
C=zeros(m,s);
```

```
for i=1:m
```

```
    for j=1:s
```

```
        C(i,j)=A(i,:)*B(:,j);
```

```
    end
```

```
end
```

```
disp('C='),disp(C)
```

- Estructura que permite repetir una serie de sentencias un **número finito** y **no preestablecido** de veces.
 - Expresión entre paréntesis en el *while*: debe tener resultado lógico (verdadero ó falso),
 - La variable/s que participa/n del *while* deben ser asignadas antes de la expresión *while* (a diferencia de *for*),
 - La variable/s que participa/n del *while* deben modificarse dentro del bloque de sentencias del *while*.
- | Pseudocódigo | MATLAB |
|-----------------------------|--------------------------|
| Do while (condición) | while (condición) |
| sentencias | sentencias |
| enddo | end |

4. Estructura tipo Mientras

- Ejemplo:
 - Desarrollar un código que calcule la norma cuadrática de un vector.

```
a=[1 2 1]
```

```
n=length(a);
```

```
suma=0;
```

```
i=1;
```

```
while i<=n
```

```
    suma=suma+a(i)^2
```

```
    i=i+1;
```

```
end
```

```
norma=suma^(1/2)
```

- Ejemplos:

$A = \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}$

$B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

$A+B$

$A-B$

$A*B$

$A.*B$

$A = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}$

$\text{inv}(A)$

$\text{det}(A)$

$[m \ n] = \text{size}(A)$

$A(2,2)$

$A(2,:)$

$A(:,2)$

Escalar por matriz (ejemplo): $2 * A$

$C = \text{ones}(3)$

$C = \text{zeros}(3)$

$C = \text{eye}(3)$

Algunas funciones de MATLAB:

<code>sqrt(x)</code>	raiz cuadrada	<code>sin(x)</code>	seno (radianes)
<code>abs(x)</code>	módulo	<code>cos(x)</code>	coseno (radianes)
<code>conj(z)</code>	complejo conjugado	<code>tan(z)</code>	tangente (radianes)
<code>real(z)</code>	parte real	<code>cotg(x)</code>	cotangente (radianes)
<code>imag(z)</code>	parte imaginaria	<code>asin(x)</code>	arcoseno
<code>exp(x)</code>	exponencial	<code>acos(x)</code>	arcocoseno
<code>log(x)</code>	logaritmo natural	<code>atan(x)</code>	arcotangente
<code>log10(x)</code>	logaritmo decimal	<code>cosh(x)</code>	cos. hiperbólico
<code>rat(x)</code>	aprox. racional	<code>sinh(x)</code>	seno hiperbólico
<code>mod(x,y)</code> <code>rem(x,y)</code>	resto de dividir x por y . Iguales si $x,y > 0$. Ver help para definición exacta	<code>tanh(x)</code>	tangente hiperbólica
<code>fix(x)</code>	Redondeo hacia 0	<code>acosh(x)</code>	arcocoseno hiperb.
<code>ceil(x)</code>	Redondeo hacia + infinito	<code>asinh(x)</code>	arcoseno hiperb.
<code>floor(x)</code>	Redondeo hacia - infinito	<code>atanh(x)</code>	arcotangente hiperb.
<code>round(x)</code>	Redondeo al entero más próximo		

Algunos operadores de relación y lógicos de MATLAB:

- Operadores de relación o comparación

Los siguientes operadores producen como resultado un valor lógico:

`true` (cualquier valor distinto de cero)

`false` (cero)

Cuando estos operadores se utilizan para comparar dos matrices de las mismas dimensiones, el resultado es otra matriz de la misma dimensión y la comparación se realiza elemento a elemento. Cuando se utilizan para comparar un escalar con una matriz, el resultado es una matriz, y se compara el escalar con cada uno de los elementos.

<code>==</code>	Igual a
<code>~=</code>	No igual a
<code><</code>	Menor que
<code>></code>	Mayor que
<code><=</code>	Menor o igual que
<code>>=</code>	Mayor o igual que

Ejemplo:

- Desarrollar un código que calcule las raíces de una función cuadrática.

En Editor cuadratica.m:

```
function cuadratica
```

```
clear
```

```
clc
```

```
a=2
```

```
b=4
```

```
c=-4
```

```
%Esta funcion retorna las raices de una ecuacion cuadratica.
```

```
%Toma 3 argumentos de entrada que son los coeficientes de  $x^2$ ,  $x$  y el término constante
```

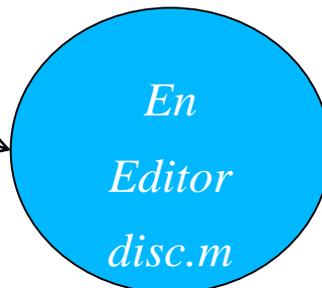
```
%Devuelve las raíces
```

```
d = disc(a,b,c);
```

```
x1 = (-b + d) / (2*a)
```

```
x2 = (-b - d) / (2*a)
```

```
end % final de cuadrática
```



```
function dis = disc(a,b,c)
```

```
%Funcion que calcula
```

```
%discriminante
```

```
dis = sqrt(b^2 - 4*a*c);
```

```
end % Final de sub-function
```

- Ejemplo:
 - Una alternativa al caso anterior sería:

En Editor cuadratica2.m:

```
function [x1,x2] = cuadratica2(a,b,c)
```

```
    function disc % función disc.m
```

```
        d = sqrt(b^2 - 4*a*c);
```

```
    end % fin de función.m
```

```
    disc;
```

```
    x1 = (-b + d) / (2*a);
```

```
    x2 = (-b - d) / (2*a);
```

```
end % fin de function cuadratica2.m
```

```
cuadratica2(2,4,-4)
```

```
ans =
```

```
0.7321
```

- Aprenda Matlab 7.0 como si estuviera en primero
 - url:
<http://ocw.uniovi.es/file.php/146/T4MaterClase/MATLAB/matlab70primero.pdf>
- Breves apuntes de Matlab - Una introducción rápida pero no trivial
 - url:
<http://personal.us.es/echevarria/documentos/IntroduccionMATLAB.pdf>
- Etc.